PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-271738

(43)Date of publication of application : 20.10.1995

(51)Int.Cl. G06F 15/16
G06F 9/46

(21)Application number : 07-041126

(22)Date of filing : 28.02.1995

(71)Applicant : NEC CORP

(72)Inventor : SURETSUSHIYU JIYAGANASAN
JIEEMUSU EFU FUIRUBIN

(30)Priority

Priority number : 94 221026 Priority date : 31.03.1994 Priority country : US

(54) CONTROL SYSTEM FOR HIGH-LEVEL PARALLEL COMPUTER SYSTEM USING SOFTWARE ARCHITECTURE

(57)Abstract:

PURPOSE: To provide the control system which uses the software architectureequipped with several layers of abstract bodiescontrolling the high-level parallel computer system.

CONSTITUTION: An abstract physical machine 10 (1st layer) includes a group of abstract physical processors and is considered as a microkernel. A 2nd layer includes a virtual machine 2 and virtual processors 16. The virtual machine is equipped with a virtual address space and a group of virtual processors

connected by virtual topology. The virtual machine is mapped to the abstract physical machines and each virtual processor is mapped to the abstract physical processors. A 3rd layer defines threads 18. The threads are processes with light weight which run on the virtual processors. The abstract physical machineabstract physical processorsvirtual machinesvirtual processorsthread groupsand threads are preferably all objects of a first class.

CLAIMS

[Claim(s)]

[Claim 1]In a control system of an advanced parallel computer system using software architecture for controlling an advanced parallel computer systemTwo or more abstract physical machines provided with two or more abstract physical processors which form one microkernel; An abstract physical processor of said plurality is accompaniedTwo or more virtual machines provided with two or more virtual processors; It has two or more thread groups provided with two or more threads which run on said two or more virtual processorsA control system of an advanced parallel computer system using software architecturewherein a virtual processor and said two or more threads of said plurality are a first-class object.

[Claim 2]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein said two or more virtual processors are connected in virtual topology.

[Claim 3]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein a user can customize a microkernel policy manager who manages a policy of said microkernel.

[Claim 4]Inside of two or more thread policy managers to whom said two or more virtual processors manage a policy of two or more of said threadsA control system of an advanced parallel computer system using the software architecture according to claim 1wherein a user includes which thread policy manager is customizable.

[Claim 5]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein said two or more threadssaid two or more virtual processorsand an abstract physical processor of said plurality cooperate functionally and build virtual topology.

[Claim 6]A control system of an advanced parallel computer system using the software architecture according to claim 5wherein a user can customize said virtual topology.

[Claim 7]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein said two or more threads allow assignment for which it could dissociate from each of those execution contextand an execution context was delayed.

[Claim 8]A control system of an advanced parallel computer system using the software architecture according to claim 1 having a port of two or more various gestalten further.

[Claim 9]A control system of an advanced parallel computer system using the software architecture according to claim 8wherein a port of various gestalten of said plurality is a first-class objectrespectively.

[Claim 10]A control system of an advanced parallel computer system using the software architecture according to claim 8wherein said two or more threads send a message containing general data and complex data.

[Claim 11]A control system of an advanced parallel computer system using the software architecture according to claim 1 to which said two or more threads are characterized for each of those local stack and heap by garbage and correcting independently of two or more of other threads.

[Claim 12]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein said two or more thread groups collect each of those share heaps independently of two or more unrelated thread groups.

[Claim 13]A control system of an advanced parallel computer system using the software architecture according to claim 1 having multiplexed said two or more

virtual processors on an abstract physical processor of said plurality.
[Claim 14]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein said two or more virtual processorsa virtual machine of said pluralityand said two or more threads exist in a durability memory.
[Claim 15]A control system of an advanced parallel computer system using the software architecture according to claim 1wherein an abstract physical processor of said plurality is a first-class object.
[Claim 16]A control system of an advanced parallel computer system using the software architecture according to claim 15wherein said two or more virtual machines are first-class objects.
[Claim 17]A control system of an advanced parallel computer system using the software architecture according to claim 16wherein an abstract physical machine and said two or more thread groups of said plurality are a first-class object.
[Claim 18]Each has a virtual processor controller and a virtual processor policy managerTwo or more abstract physical processors connected in physical topology and; of each. They are two or more virtual machines which have virtual address space and two or more virtual processorsand computer systems provided with;Said two or more virtual processors of each of two or more of said virtual machinesAnswer said virtual processor controller and said virtual processor policy managerand it performsAnd it has a thread controller and a thread policy managerSaid two or more virtual processors are connected in virtual topologyand each virtual processor is mapped by each abstract physical processorand said computer systemsComputer systems having further two or more threads which run on said two or more virtual processors which answer said thread controller and said thread policy manager.
[Claim 19]The computer systems according to claim 18 having multiplexed said two or more virtual processors on an abstract physical processor of said plurality.
[Claim 20]The computer systems according to claim 18wherein it has a durability memory and an object containing said two or more threadsa virtual processor of

said pluralityand said two or more virtual machines exists in this durability memory.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application]This invention receives a present-day programming language especially about the control system of the advanced parallel computer system using the computer software architecture for controlling the computer systems parallelized highlyIt is related with the control system of the advanced parallel computer system using the computer software architecture designed serve as a very efficient substrate.

[0002]The control system using this computer software architecture is based on the operating system which separated the problem of control from the problem of the policy. Two different abstract object levels of a system are performing this separation. that isit comes out in an abstract physical processor and a virtual processor. Each of these abstract object is divided into two components. One is a "controller" which realizes the control section of an abstract object.
Another is a "policy manager" who determines a policy to a controller.
By separating control from a policyit becomes possible to perform the definition of different **** to the same system only by changing the policy manager portion of an abstract object functionally.

[0003]Specificallythe control system using this software architecture supports the lightweight thread and virtual processor of control as a first-class object. A first-class procedure and first-class continuation realize parallel (KONKARENSHI) management. Even if the user does not have the knowledge about a fundamental runtime system by itit becomes possible to optimize **** of the run time of application.

[0004]This invention still more specifically relates to a design with realization of a thread controllersystematization of large-scale parallel calculationand the firm programming environment for parallel computingusing a continuation as construction of asynchronous parallel structureand a fundamental control mechanism.

[0005]

[Description of the Prior Art]In order for the interest over parallel computing to increase and to express parallelism as a resultmuch parallel programming languages which define the program of a high level and the structure of data clearly were produced. The parallel language which targets a nonnumerical application area supports typically generation of a dynamic lightweight processa high-level synchronous unmodified instructiondistribution data structureand the parallel structure of realizing speculative parallelism (efficiency differs). It is thought that such parallel languages comprise two sub-languages altogether as a matter of fact. It is with two sub-languagesi.e.the adjustment language which manages and synchronizes the activity of the meeting of a processand the calculation language treating the data object limited to the given process.

[0006]Traditionallythere are some classes in an operating system. For examplethey are real timeinteractive (interactive mode)a batchetc. Since the operating system of these three classes provided a different interface to a userit was difficult to move a program to the operating system of other classes from the operating system (OS) of a certain class. Since the scheduling for which the operating system of each class opts differsIt is difficult to debug the program (for examplereal time application) for one operating system with other operating systems (for exampleinteractive mode development system)It is difficult for application to have confidence about running correctly and efficiently on a target system.

[0007]And since a scheduling system which is different in the system of these classesrespectively is usedthe situation is still more complicated. For exampleto two or more processes [real-time system / a certain kind of]an order of

scheduling uses priority order in another system to fixingor has combined them in the system of further others using the running quantum. The interactive mode operating system or the batch operating system has most number of choices about scheduling.

[0008]By separating control from a policyan operating system easily customizable to the operating system of various classes can be built. The module which realizes a policy manager in this invention is dramatically small compared with the size of a system typically. Generally the line number of a code is less than 100. Thereforewhen newly building the system by which **** of a policy differthe small part of a code may usually only be written. Since a policy manager provides the interface defined wellwhen **** of a policy is changedhe does not need to examine the whole system and should examine only a new policy manager.

[0009]Hydra (reference: "HYDRA/C.mmpi:An Experimental Computer System"William WulfRoy Lexia and Samuel Harbison workMcGraw-Hill1991)It is the first operating system that meant separation with control and a policy and was designed. HoweverHydra accepts customization of the policy only on Carnell's level. It advances furtherand when they are the things relevant to a specific programit enables it to customize the determination of a policy in this invention. Thereforeconversational programssuch as an editor and a window managercan have a dramatically different policy from the program which does a subject calculationssuch as a simulation of hydrodynamicsand calculation of the finite element method. Separation with the control and the policy in Hydra has required cost. That is because two or more context switches are needed between Carnell and a policy manager. Generally in this inventionthe policy manager links to the suitable address space directlyand the change of a context is unnecessary. Thereforethe policy manager of this invention is efficient at least to the same extent as the policy management (it is not customizable) in the conventional operating systemand usually more efficient than before.

[0010]One method of realizing the parallel language of a high level is building a

virtual machine (user levels) for exclusive use. A virtual machine functions as a substrate which realizes fundamentally the parallel primitive of a high level looked at by the adjusting part language. When the adjustment language L supports the parallel primitive Pthe role of the virtual machine (L_P) of L is treating all the things in relation to realization of P. For that purposeit is often necessary for a machine to manage the scheduling of a processmemorymanagementsynchronizationand others. Howeversince it is accepted and L_P is adjusted so that P may be realized efficientlyin many casesit is not appropriate to realize a dramatically different parallel primitive. Thereforein order to build the dialect of L by parallel primitive P'it is necessary to newly build a virtual machine or to usually express semantic regulation of P' using P. These two approaches have a fault clearly. That isit is a high cost in order to newly build a complicated virtual machine in the 1st approach. On the other handthe semantic regulation of P of the 2nd approach is a high level.
Since the function of L_P is limitedit is insufficient.

[0011]In realization of languagethe function of the operating system of a low can be used instead of building a virtual machine for exclusive usein order to realize parallelism. The thread (a heavyweight or lightweight) of control which OS manages realizes generation and scheduling of a process. And synchronization is treated using the structure which OS of a low manages. Thusgenerallyit is portable and what was realized has extendibility higher than the system built around the runtime system for exclusive use. Howeversince all kernels (low) need to cross the protection boundary between application and an operating systemefficiency falls victim. since general-purpose OS function is usually a feeling of un-to semantic regulation of the target parallel operator -- them -- the point of compile duration or execution time -- optimization -- almost -- or it does not carry out at all.
[0012]
[Problem(s) to be Solved by the Invention]In order to control an advanced parallel

multiprocessor / multi-computer systemThe control system of the advanced parallel computer system using the operating system architecture of the computer which serves as an efficient substrate very much to a present-day programming language is obtained. According to this inventionthe control system of the advanced parallel computer system using the software architecture for the asynchronous calculation based on a customizable virtual machine is obtained.
[0013]According to this inventionthe control system of the advanced parallel computer system using the software architecture which supports a lightweight thread as a first-class object on a virtual processor is obtained.
[0014]According to this inventionthe control system of the advanced parallel computer system using the software architecture which contains especially in user levels the policy manager who can customize is obtained.
[0015]According to this inventionthe control system of the advanced parallel computer system using the software architecture containing customizable virtual topology is obtained.
[0016]According to this inventionthe control system of the advanced parallel computer system using the software architecture containing the thread groups as a place thread absorptiondelay TCB assignmentand memory storage shared is obtained.
[0017]According to this inventionthe control system of the advanced parallel computer system using software architecture including the port of various gestalten is obtained.
[0018]According to this inventionthe computer systems controlled using the above software architecture are obtained.
[0019]
[Means for Solving the Problem]In a control system of an advanced parallel computer system using software architecture for controlling an advanced parallel computer system according to this inventionTwo or more abstract physical machines provided with two or more abstract physical processors which form one microkernel; An abstract physical processor of said plurality is accompaniedTwo

or more virtual machines provided with two or more virtual processors; It has two or more thread groups provided with two or more threads which run on said two or more virtual processorsA control system of an advanced parallel computer system using software architecturewherein a virtual processor and said two or more threads of said plurality are a first-class object is obtained.

[0020]According to this inventioneach has a virtual processor controller and a virtual processor policy managerTwo or more abstract physical processors connected in physical topology and; of each. They are two or more virtual machines which have virtual address space and two or more virtual processorsand computer systems provided with;Said two or more virtual processors of each of two or more of said virtual machinesAnswer said virtual processor controller and said virtual processor policy managerand it performsAnd it has a thread controller and a thread policy managerSaid two or more virtual processors are connected in virtual topologyand each virtual processor is mapped by each abstract physical processorand said computer systemsComputer systems having further two or more threads which run on said two or more virtual processors which answer said thread controller and said thread policy manager are obtained.

[0021]This invention relates to realization of an adjustment substrate which makes it possible to express a parallel structure of a wide range in a context of a high-level programming language. This invention defines a general-purpose adjustment modeland enables it to realize much special adjustment languages efficiently on the model. in operation of this invention -- scheme (Scheme) (reference: -- "The Revised Report on the Algorithmic Language Scheme"ACM Sigplan Notices and 21 (12).) of software It used as the foundation of calculation of 1986 and Jonathan Rees and Wiliam Clinger. A scheme is a dialect of Lisp when [higher order] it sees in dictionary. Although a scheme is a desirable languagea design of the above-mentioned adjustment substrate can be taken in to a programming language (high level) of any present age so that clearly for a person skilled in the art.

[0022]. An operating system of this invention used a shared memory or a separation memory fundamentally. It is designed run on a distributed machine which is designed run on an MIMD (multi-instruction multi data) parallel computerand comprises a network of a workstation. In software architecture of this inventionin performing on a machine which used a separation memory or a dispersion memoryit uses a share virtual memory model. In the realizationan algorithm with which different a large number corresponding to a parallel paradigm differ was used. Result parallela master / slave paralleland logical parallel are included in the above-mentioned parallel. Some different parallel programming models were realized on an operating system provided with a futurefirst-class tuple (group) spaceand an engine.

[0023]A dialect (it is called the sting (Sting)) of a scheme which constitutes an operating system (OS) which is the feature of a desirable example of this inventionIt has combined a best point of two approaches including an adjustment language (an exclusive virtual machine realizes) for expressing asynchronous and lightweight parallelism. Unlike a system of other parallel schemesand a parallel dialect of a high level language of the same kinda fundamental concurrent object (a threada virtual processorand a physical processor) in a sting is the data structure formed into the stream lineand does not need complicated synchronization. A parallel system which depends for management of parallelism on service by OS is differed fromA sting becomes possible [optimizing **** of run time of application]without an object and a procedure of a scheme realizing a problem of all the parallel managementsand as a result a user having the knowledge about service of OS in back. A sting is the framework simplified notionally and supports the fundamental feature for generating and managing the asynchronous parallelism of various gestalten according to a very general framework. It turned out that an efficient substrate which can build various parallel dialects of a high level language on it as a result can be built. It is not what meant that a sting only considered it as a medium means to realize a program of a short life of a stand-aloneand expects to provide a framework for

building a rich programming environment for parallel computing. Thereforethis system supports a scheduling policy depending on treatment and application of an exception beyond thread PURIEMPUSHONan asynchronous garbage collection for every threadand a thread boundary. This system has a required functionin order to treat a lasting long lasting objectmultiple address spaceand the other features common to the newest programming environment.

[0024]In a stinga virtual processor is multiplexed on an abstract physical processorand a thread is multiplexed on a virtual processor. All decision of a policy relevant to this multiplexing is made by policy manager. A decision about multiplexing of a thread on a virtual processor which makes a decision relevant to multiplexing of a virtual processor on a physical processor by a virtual processor policy manager (VPPM) is made by thread policy manager (TPM).

[0025]A policy manager determines three types. Namely. [how when an object is generated or resumeda new object (VP or thread) is mapped in a processor (physical or imagination)and] It is remapping or three of whether to move when at a processor of others [processor / a certain] about an order made to run two or more objects mapped by specific processor and an object.

[0026]A sting is the operating system designed support a programming language of the present agesuch as a schemeSmalltalk (Small Talk)MLMOJURA 3 (Modula3)or Haskell (Haskell). A sting gives the foundation of a rectangular building body of a lowand enables a designer or a user of language to build simply and efficiently various building bodies which the above-mentioned language needs by it.

[0027]A present-day programming language requires many more compared with programming languagessuch as the conventional COBOLFORTRANCor Pascal. Although a sting is designed support a present-day programming languagethe conventional programming language is supported efficiently similarly. A present-day programming language lists a different point from the conventional language below.

[0028]- Parallelism : it is easy to use a general-purpose multiprocessor

increasinglyit has becomeandas a resultis [for a parallel programming] efficientand interest is increasing to construction of a plat form excellent in power of expression. Efforts for for including parallelism in a programming language of a high level are paid to a point that most adds a unmodified instruction of a special purpose to language.

[0029]- Multi-synchronization model : many synchronization protocols are used in a parallel programming or asynchronous programming. The present-day operating environment must provide a unmodified instruction which supports most various possible protocols.

[0030]- Lazy (delay) evaluation and Eager evaluation : many present-day languages are supporting either lazy evaluation or the Eager evaluation and both. Lazy or it is important for an operating system to prepare even Eager's perfect evaluation strategy.

[0031]- Automatic storage and file management : this has been the fundamental feature of much present-day languages. That is because a program is further made to a thing excellent in power of expressionand an error of a program is reduced simultaneously and complexity of a program can be eased by automatic storage and file management.

[0032]- Topology mapping : although not supported yet in many programming languagesso that a communication overhead in a program may be reducedThe capability to control mapping to a processor of processing will become more importantif size of a multiprocessor computer system continues becoming large and topology becomes more complicated at all.

[0033]A sting supports an element of these versatility efficiently. In a much more general and more efficient architecture frameworka sting performs this from what can be used now. A sting provides a program with high power of expression and a controllabilityand customization ability of an un-parallel level again.

[0034]A sting can be best distinguished from other parallel languages according to four features in the design.

[0035]1. Parallel abstract object : parallelism is expressed by lightweight thread

of control by a sting. A thread is an un-strict first-class data structure.
[0036]2. A processor abstract object and a policy abstract object : perform a thread on a virtual processor (VP) showing an abstract object of schedulingand load balancing and a protocol. There may be more virtual processors than the number of physical processors which can actually be used. Like a threada virtual processor is a first-class object. One VP is provided with one thread policy managerand this policy manager determines scheduling and a shift method for a thread which it performs. Different VP can be actually provided without a fall of **** with a different policy manager. A virtual processor is performed on a physical processor which is a actual physical computing device.
[0037]a meeting and an address space of a virtual processor -- ****** -- one virtual machine is formed. It can perform on a physical machine with two or more single virtual machines. A physical machine comprises 1 set of physical processors. It is an object of a scheme which can also direct a virtual machine and a physical machineand is operational as this object.
[0038]3. Memory model : one thread assigns data to a stack and a heap which the thread manages exclusively. Thereforetwo or more threads perform a garbage collection of those private State mutually-independent. Global synchronization is unnecessary when a certain thread puts a private garbage collection into operation. Cross and refer to the thread for data. When performing a garbage collection of an object across a thread boundaryreference information between fields is used. Memory is managed by a generation scavenging collector. Long lasting data assigned by one thread or self-sustaining data can access other threads in the same virtual machine.
[0039]A design of this invention is considered at a locality of memory. For examplecash of the memory storage for making a thread run is carried out to VPand it is recycled when one thread is completedand so that it can reuse immediately. Two or more threads can always share the same execution contextwhen the dependency of data is guaranteed.
[0040]Programming model : 4. A sting permits an exception which should be

treated across boundaries between threadsSupport non-blocking I/O and customization of scheduling of a virtual processorAs can customize scheduling of a thread on a virtual processorit supposes that it is possible and an internal structure which realizes multiple address space and a share self-sustaining object is given. A sting supports delivery with sufficient efficiency of a message which used a port of first-class various gestalten again. A port is useful to ease an overhead in realization of a shared memory on a separation memory plat form.

[0041]In software architecture which controls an advanced parallel computer system of this inventionan operating system (sting)a base languageand a compiler are unified in one abstract machine. Start points are high-level programming languagessuch as a scheme. This programming language is expanded with an efficient abstract object including a threada virtual processorand a policy manager. This outstanding operating system includes a mechanism which used effectively a trend of the present architecture of attaching a premium to a locality of data.

[0042]As a resulta mechanism which builds an efficient adjust construction object for parallel computing was acquired. The foundation of a progressive programming environment is acquired by using a lightweight thread. An efficient asynchronous system is obtained by supporting a locality of data.

[0043]It is a concept of virtual topology that it is central for performance of this system. Virtual topology defines a relation to a meeting of a virtual processor. A treea grapha Hypercubeand processor topology constituted as a mesh are the example known well. Virtual processors are scheduling and migration which a thread performsand an abstract object which defines a policy of load balancing. It has intention of this virtual topology so that a framework of a high level which defines mapping (details of a low of physical interconnection are abstracted) of a complicated thread and a processor and which was simple and was excellent in power of expression may be given.

[0044]A thread generated by calculation is mapped to a processor in virtual topology by mapping function relevant to the topology. The user can define these

mapping functions. When a system is realized on a specific multiprocessor plat form using virtual topologyit is possible to define a procedure which maps a virtual processor in virtual topology in a physical processor in a plat form.

[0045]Unless it includes reference to interconnection of a physical processor or a physical processor in itself [code]with a machineit is independently. In the method of passage of a node in specification of virtual topology which a program usesand topology at the time of program executionall the things about thread mapping and a locality are abstracted.

[0046]Profits of virtual topology and processor mapping are not only in efficiency but in a point of portabilityand it becomes unnecessary to specialize realization of a parallel algorithm for every individual physical topology by it. Since a mapping algorithm which relates a thread with a processor is finely specified as a part of virtual topologythe programmer can manage correctly how a thread should be mapped to a virtual processor. When it turns out that communication is needed in a certain calculationthe ability to assign these threads clearly to a specific virtual processor can perform load balancing superior to a case of an implicit mapping strategy. Structure of a graph of control defined by parallel algorithm and data flow can be used in various forms. When a meeting of a thread is sharing common datait is possible to build topology which maps a virtual processor which these threads perform in the same physical processor. A thread multiplexes a virtual processor the same with multiplexing on a virtual processor on a physical processor. When a meeting of a certain thread needs important mutual communicationtopology which maps those threads in a processor mutually near in virtual topology can be built. As for T_1 and $T_2$when thread T_1 has data dependency to a value which other thread T_2 generatesmapping in the same virtual processor is rational. An opportunity to improve granularity of a thread is given with capability for a processor to be able to perform scheduling to a data dependency thread on a same or near processor in a fine program of granularity which will almost be in a busy state. Finallya certain kind of algorithmssuch as an adaptation tree algorithmhave the process structure of developing along with

advance of calculation. On topology in which dynamic generation of a virtual processor is possiblethese algorithms are performed best.

[0047]There is a role of a continuation and a first-class procedure in an operating system of a general-purpose multithread efficient as an outstanding field of everything but this software architecture and realization of program environment. A continuation is used in order to realize operation of a change statetreatment of an exceptionand optimization of important memory. A continuation is an abstract object of a program point. A continuation was expressed by procedure which has one argument and this procedure defines the remaining calculation that should be performed from a program point which an argument shows.

[0048]Virtual address space of a sting is constituted by 1 set of fields. A field is used in order to systematize data in which a strong locality is shown temporarily or spatially. A sting supports various fields. That isthey are thread control blocka stacka thread private heapa thread share heapetc. A field which data is assigned to a field based on those specifications and lives that were meanttherefore is different will be provided with a different garbage collector relevant to them.

[0049]An exception and interruption are always treated in an execution context of a certain thread like [in the case of a context switch of a thread level]. An exception handler is realized by procedure of the usual schemeand exceptional dispatch includes operation of a continuation fundamentally.

[0050]Some characteristics are shared with a thread package system developed for other high level languages as long as a sting is a programming system which manages [generation and] control. [of a lightweight thread] These systems also regard a thread as a clear data typeand PURIEMPUSHON is supported to various gradesand a programmer is enabled to specify special schedule management in a certain limited case. In these systemsan abstract object of a thread has defined an adjusting part language.

[0051]Howevera sting differs from these systems at some important points. A protocol of scheduling and migration which a sting uses [1st] is thoroughly customizable. Different application can make a different scheduler runwithout

changing Thread Manager or an abstract object of a virtual processor. Such customization is applicable to a virtual machine's own systematization. Oppression of a process by a support of a locality of data based on a stingoptimization of memoryand absorption of a thread cannot be performed [2nd] in other systems. All operations of a thread are directly realized within a virtual machine of a thread. There is no context switch to a kernel of a low which should be carried out for execution of operation of a thread. A sting is built in an abstract machine which meant supporting long lasting applicationa lasting objectand multiple address space. Since a thread package has not defined program environment with perfect (definition) themthese functions are not provided at all.

[0052]Since a sting is designed as a system-programming languagea parallel abstract object of a low is provided. The application library can generate a thread object directlyand can define scheduling and a thread migration strategy of these selves. Although a parallel building body of a high level is realizable using a threadhowever if efficiency is guaranteeda system will not forbid a user from using operation of a thread directly as mentioned above. In environment at the time of the same executionthe same application is different efficiency from different semantic regulationandspecificallycan define a parallel abstract object.

[0053]A sting resembles other progressive multithread operating systems at a certain point. For examplea sting is supporting excess interruption which a call-back and a user manageand a non-blocking I/O call accompanying management of local address space as operation of user levels. A sting divides a matter of user levelsand a matter of a kernel level. A physical processor treats operation of a system (a privilege was able to be granted)and operation over two or more virtual machines. A virtual processor realizes all functions of a thread of user levelsand local address space. Howeversince a sting is an extended dialect of a schemeit provides functionality and expression nature of a programming language of a high level which are not provided in a typical operating system environment.

[0054]A sting builds an asynchronous programming primitive and is a plat form for experimenting in a paradigm of a new parallel programming. It is possible to evaluate the technique of different parallelism by the design in race. Semantic regulation is defined welland simply on the wholesince it is efficienta scheme provides especially rich environment for conducting such an experiment. Howeverit does not depend for a design of a sting on language in itself. Thereforeit can include in any high-level programming languages very easily.
[0055]A hook is not given to each parallel paradigm and each parallel primitive which are considered for a sting to be only interesting. It has focused on basic structure and a function common to a parallel programming structure that that is not right and wide range. Thereforerealization of blocking is easily usedin order to support logical calculation. Optimization of thread absorption used in order to deter execution of a threadIt is dramatically suitable for realizing synchronization of a future and tuple spaceand a policy manager who can customize makes it possible to build a fair and efficient scheduler to other various paradigms at the end.
[0056]
[Example]Nextthe example of this invention is described with reference to drawings.
[0057]The block diagram of the control system of the advanced parallel computer system using the software architecture for controlling the advanced parallel computer system by one example of this invention is shown in drawing 1.
[0058]abstraction -- physical -- a machine -- (-- PM --) -- ten -- physical -- topology -- (-- PT --) -- 11 -- mutual -- connecting -- having had -- abstraction -- physical -- a processor -- (-- PP --) -- 12 -- constituting -- having -- **** . This abstract physical machine is used in order to perform 1 set of virtual machines (VM) 14. it -- receiving -- each -- a virtual machine -- imagination -- topology -- (-- VT --) -- 20 -- 20 -- ' -- connecting -- having had -- one -- a ** -- more than -- a virtual processor -- (-- VP --) -- 16 -- having -- **** . The thread (T) 18 is performed on one or more virtual processors in the same virtual machine. The

specific thread can shift between different virtual processors in the same virtual machine 14 (my great). The thread policy manager (TPM) 19 (shown in drawing 2 and drawing 3) controls the scheduling of a threadand load balancing and the policy of a thread. The relation between different elements and the details of each element are explained below.

[0059]Software architecture (it may be called the operating system architecture) can be considered to be the arrangement of the layer of some abstract objects (drawing 2). The 1st layer contains the abstract physical machine 10and this machine contains the group of the abstract physical processor 12. This layer supports what is called micro kernel in the present operating system. The following layer contains the virtual machine 14 and the virtual processor 16. The virtual machine is provided with virtual address space and the group of the virtual processor connected by virtual topology. A virtual machine is mapped by the abstract physical machine and each virtual processor is mapped by the abstract physical processor in that case. The 3rd layer of an abstract object is the thread 18. These threads are the lightweight processes of running on a virtual processor.

[0060]Virtual topology is a tree of the virtual processor mapped by the physical processor physically connected by mesh topologyfor example. By virtual topologya programmer becomes possible [expressing a program by topology (imagination) suitable for the algorithm which should be carried out]. A sting provides efficient mapping to the actual physical topology of a target machine from virtual topology. It becomes possible to move a parallel program easily between different physical topology by virtual topology.

[0061]The main components of the adjusting part language of a sting are a lightweight thread and a virtual processor. A thread is a simple data structure including a local memory unit (namelya registera stackand a heap)a codeand related state information (namelystatusa prioritya PURIEMPUSHON bita locketc.). They define the place of the independent control. This system does not impose restrictions to the code which a thread contains. All expressions of an effective scheme are treated as an independent process.

[0062]As shown in drawing 2 and drawing 3as for each virtual processor (VP) 16this controller carries out a state transition function on a thread and the thread policy manager (TPM) 19 including the thread controller (TC) 17. And a thread policy manager carries out both the scheduling of a threadand load balancing / shift policy. Although each VP shares a thread controller within the same virtual machinedifferent VP can have a different thread policy manager.
[0063]The thread 18 has multiplexed the virtual processor 16 the same with having multiplexed on a virtual processor on the physical processor 12. Each physical processor supports the calculation engine in multiprocessor environment. The virtual processor controller 13 and the virtual processor policy manager 15 relate to each physical processor PP. A virtual processor controller performs a context switch between virtual processors by PURIEMPUSHON or an explicit request. A virtual processor policy manager treats the determination of the scheduling to the virtual processor 16 performed on the physical processor PP. For examplethe virtual processor can abandon control of a physical processorwhen the thread is not performing on itand when a thread cannot be transferred from other VP. The physical processor can make the virtual processor of any virtual machines in a system run.
[0064]The related virtual processor can access it exclusively including the address space 24 where a virtual machine is single. The information in the global storage pool 26 with a global virtual machine. (For examplea librarya file systemetc.) can be sharedand the global common object 28 (namelyobject in global address space) is mapped to those local address space. The virtual machine contains the route of the activity object graph (namelyroute environment 30) used again in order to trace all the activity objects in an address space.
[0065]All the sting objects (a threadVPand a virtual machine are included) exist in a durability memory. This memory is constituted as a set of a separate area. An object is the garbage collected in the field using the generation collector. Refer to all other objects in the address space for one object. At firstan object exists in the thread local field of a short life. In a generation hierarchyit moves from the object

surviving from the garbage collection to a higher rank. This function is completely clear for a user.

[0066]A first-class object is an object which can be passed as an argument to a procedurecan be returned from a procedure as a resultor can be memorized in a data structure. In the desirable example of the abstract physical machine of this inventionall of an abstract physical processora virtual machinea virtual processorthe group of a threadand a thread are first-class objects. In other examplesonly a thread and a virtual processor are first-class objects.

[0067]A sting compiler is an improvement version of Ovid (Orbit). Ovid is described by the paper of D.Kranz and others (July reference: "Orbit:An Optimizing Compiler for Scheme"in ACM SIGPLAN Notices21(7):219-233). 1986. The target machine which appears with a compiler contains the thread register holding the reference to the thread object which is running now for exclusive use. The severe operation of evacuation and time restrictions of restoring or assigning the storage area (namelya stack and a heap) of a thread is prepared as basic operation on a context switch in a register. A continuous scheme program is compiled without change and executed. In the stinga futuredistribution data structureand speculative parallel operation are also realized. A scheme program can be made to expand freely by the parallel operation supported by either of these paradigms.

[0068]A thread is a first-class object in a sting. Thereforethey can be passed to a procedure as an argumentand can be returned as a resultand can be further stored in a data structure. A thread can be survived to the object which generated the thread for a long time. The state of the thread contains the NURARI (nullary) procedure exercised when the thunk (thunk)i.e.a threadis performed. The value of application is stored in a thread at the time of an end. For examplethe lightweight thread of the control which exercises a thunk (lambda() (+y (*xz))) is generated by evaluating the expression (fork-thread (+y (*xz))). The evaluation environment of this thunk is the dictionary environment of expression called fork-thread.

[0069]A thread records state information as a part of the state (refer to drawing 4 and drawing 5). A thread takes the state of the delay 36the schedule 38the evaluation 40the absorption 42 or the decision 44 either. The delayed thread is not run unless the value of a thread is required clearly. Although the scheduled thread is a thread which one of VP knowsthe storage resource has not been assigned yet. The thread which is evaluating is a thread which began to run. A thread stops at this state until the application of that thunk achieves results. The state of a thread is become final and conclusive when the above-mentioned result comes out. The absorbed thread turns speciallyand since it is importantit explains the thread under evaluation in more detail below.

[0070]In addition to state information and the code which should be evaluatedone thread includes again the system information which the reference information to other threads which are waiting to complete itand (1) (2) thunk are dynamicand contains exceptional environmentthe parents of a threada brotherand a child.

[0071]each thread is also used in order to realize fluid (that isdynamic) combination and treatment of an exception -- it is dynamic and has exceptional environment. System information is useful as a tool of debugging and profilingand application becomes possible [monitoring dynamic deployment of a process tree] by it.

[0072]In realization of a threadit is not necessary to change other basic operations in language. The synchronization semantic rule of a thread makes the synchronization function which can be usedfor example by "touch" of MultiLispthe tuple space of Lindaand "sync" of CML a more general (even if it is a low) form.

[0073]The application can control a state thoroughly and the thread which is a basis of the state and was blocked can be revived. Howeverthere is an explicit system support to data flow (namelyfuture touch)non-deterministic selection and the synchronization based on restrictionsor barrier synchronization.

[0074]A user operates a thread by 1 set of procedures (it lists below) which a thread controller (TC) and (transition of a synchronous state is realized in the

state where there is a thread) define. As for TCit is desirable to write the whole by a scheme except for two basic operations of evacuation and restoration of a register. A thread controller does not assign a storage area. Thereforethe call of TC does not carry out the trigger of the garbage collection. In addition to such operationsthe thread can go into a controller for PURIEMPUSHON. A thread procedure is shown below.

[0075](fork-thread expr vp) generates a threadin order to evaluate exprand it schedules it so that it may be run on vp.

[0076](dealy-thread expr) generates the delayed thread which evaluates exprwhen required (thread value).

[0077](thread-run thread vp) inserts delayed thread which was blocked or suspended in lady queuing of vp.

[0078]The thread which is performing this operation is made to block (thread-wait thread) until the state of thread is become final and conclusive.

[0079](thread-block thread .blocker) requests that it blocks in thread. blocker(s) are conditions in case a thread blocks.

[0080](thread-suspend thread .quantum) requests suspension of execution to a thread. A thread is resumed when a quantum argument is givenand the specified period passes. When that is not righta thread is indefinitely suspended until it is clearly resumed using thread-run.

[0081](thread-terminate thread .values) requests that values is ended as the result to thread. (yield-processor) requests the present thread to stop control of the VP. This thread is inserted in suitable lady queuing.

[0082](current-thread) returns the thread which is performing this operation.

[0083](current-virtual-processor) returns the virtual processor by which this operation is evaluated on it.

[0084]In order that a user may explain how a thread is programmablethe program of drawing 6 is considered. This program defines realization of an easy prime number discovery means. Any specific parallel paradigms are not referred to in this definition. Such a problem is abstracted by the op argument.

[0085]It depends for realization of this prime number discovery means on the synchronous thread abstract object which gives the blocking operation (hd) in stream accessand the atomic operation (attach) added to the last of a thread and which the user defined.

[0086]Processing of the versatility of a prime number discovery means by which the grades of asynchronous **** differ can be defined. For example (let (filter-list (list)))

(sieve(lambda(thunk

(set filter-list(cons(delay-thread(thunk))

(filter-list))))))

A filter is generated lazily. Once a filter is requiredit will remove an element from an input stream repetitivelyand will generate a potential prime number on an output stream. Since the new filter scheduled on VP using round robin thread arrangement order is startedit can write as follows.

[0087]

(thread-run(car filer-list)

(mod(1+(vm.vp-vector(vp.vm(current-virtual-processor))))

n))

The expression (vp.vm (current-virtual-processor)) defines the virtual machine which makes present VP a part. The public State of the virtual machine includes the vector which accommodates the virtual processor.

[0088]By rewriting a little above-mentioned call to a sheaverealization of a lazier prime number discovery means can be expressed.

[0089]

(let((filter-list(list)))

(sieve(lambdas(thunk)

(set filter-list (cons(create-thread (begin (map thread-run filter list)

(thunk)))

filter-list))

(map thread-block filter-list))

n))
In this definitionthe filter which encountered the potential prime number p generates the lazy thread object Land requests that it blocks in other filters of all the in a chain. When the value of L is requireda filter unlocks all the elements in a chainand removes all the multiples of p in the input thread. In this callextension of a sheave and consumption of an input are controlled based on a demand.
[0090]This sheave is also changeable into an as follows more Eager version.
[0091]
(sieve(lambda(thunk
(fork-thread-(thunk)(current-vp)))
n)
By evaluating this applicationthe new thread for removing all the multiples of a prime number is scheduled. This thread is scheduled on the virtual processor which performs this operation. In this callwhenever a prime number is newly foundthe thread to evaluate is generated.
[0092]In a stingthe operation of a thread is treated as a usual procedure and the object referred to by the operation of a thread is operated as the object besides either of a scheme. When two filters tied by the common stream are completedthe storage area which the above-mentioned stream occupies can be reused. A sting does not impose a transcendental synchronization protocol to access of a thread. He is trying for an application program to build the abstract object which prepares adjustment of a thread.
[0093]The thread generated with the filter is ended by one in two methods. The call of the top level to a sheave can be constituted so that it may have an explicit handle to these threads. The filter list data structure used in order to generate a lazy sheave is the example. To the next (map thread-terminate filter list)
It can evaluate and all the threads in a sheave can be terminated. Or the application can manage these threads collectively using the group of a thread.
[0094]A <thread groups> sting gives thread groups as a means to gain the control to the meeting of a related thread. One thread groups are generated by

the call to fork-thread-group. This operation generates a new group and a new threadand a new thread turns into a new group's route thread. A child thread shares the same group as the parentsunless a new group is generated clearly. As for one groupall those members can access this heap including one share heap. I would like to end thread groups by the next call (thread-group-terminate group).

All the valid threads in a group are ended and the garbage correction of the share heap is done.

[0095]The debugging operation and thread operation which put together it all and can apply it also contain thread groups to the member again. Thread groups with the operations (for examplereport etc. of the listing of all the threads in the given groupall the groups' listingprofilingand a system) for debugging and a monitor. The operations (for examplean endsuspensionetc.) of the usual thread and operation of the same kind are provided. Thereforewhen the thread T is completedthe user can request as follows to all the children (it defines as some groups of T which should be ended) of T.

[0096](thread-group-terminate(thread.group T))

Thread groups are the important tools for controlling a share in hierarchical memory architectures. Since the object which a group's member shares is contained in a group's share heapas for these objectsit is desirable that it is mutually close physically within a memoryand a better locality is obtained by it. Thread groups can also be used as a place of scheduling. For exampleunless a thread policy manager is permitted that all the threads in a group runeach thread in a group can realize the scheduling policy that it cannot run. This scheduling system is as of the same kind as a "gang scheduling" protocol. Thread groups can be used with virtual topologyin order to improve the locality of data.

[0097]When a <execution context and thread control-block> thread starts evaluationan execution context is assigned to it. Each thread which is evaluating is connected with the execution context known also as the thread control block (TCB) 32 (drawing 5). TCB generally expresses a continuation and contains the

stack 31 and the local heap 33 of itself. Both a stack and a heap can be restrained and the garbage collection of the heap is carried out using a generation scavenging collector. The values of the lock in which TCB is related in addition to a memory objectand all the useful registers that remain when a thread finally performs a context switchThe substate (for examplestatessuch as initializationa ladyevaluationa blockand suspension) of a thread and the thread contain VP performed at the endthe priority of a threadand the time quantum.

[0098]The transition diagram of the thread State and a thread substate is shown in drawing 4. The state of TCB is reflecting the operation permitted on the thread which is evaluating. If the thread T under evaluation has TCB $T_{TCB}$the State field of T_{TCB} shows any one of the following inside.

[0099]Initialization 46: Although the stack and heap relevant to T_{TCB} are initializedno code is executed yet.

[0100]Lady 48: Although T can be performed on what kind of VP which can be usedit has not performed yet on which VP now.

[0101]Run 50: T is performed on a certain VP now.

[0102]Block 52: T is blocked on a certain thread or under some conditions now.

[0103]Suspension 54: T is suspended indefinitely fundamentally.

[0104]End 56: T ended execution and has swept away the remaining states.

[0105]It is not an object which is [as for TCB] visible to a first-class user unlike a thread. Only a thread controller and a thread policy manager can access them. TCB is assigned to it when a new thread is in the ready state of a run. When a thread changes into a definite statethe thread controller can use the TCB for the thread generated behind. TCB does not run in in the data structure which a user maintains. TCB is exclusively operated by the procedure of a system level.

[0106]Realization of a sting is postponed until it is needed in the assignment of a storage area to a thread. The operation which generates a thread not only sets up the environment to the thread which should only be forkedbut in other thread packagesassignment and initialization of the storage area also include it. In this approachdecline in efficiency is caused with two important points. Longer time is

consumed to generate and initialize [1st] them from a thread controller making a thread actually run by the fine parallel basis of granularity. Since a stack and a process control block are assigned [2nd] shortly after a thread is generatedas for the context switch between threadsthe advantage of the locality of cash and a page is often unutilizable. When assignment of TCB is not delayedall the memory space required for a system will increase substantially.

[0107]Thread control block of a sting is recyclable resources managed by the virtual processor. TCB is assigned to a threadonly when a thread starts evaluation. The strategy of this assignment is designed improve the locality of data. TCB can be assigned by one in four methods to the thread T which should run on VP V.

[0108]1. When the thread under execution is completed on the present Vthe context can be promptly used for re-assignment. The TCB is the best candidate for assignment. It is because this TCB has the highest locality to that VP. The physical cash and the memory relevant to this VP have highest possibility that the execution context of the thread which ran on VP most recently is included.

[0109]2. When the thread performed now is not completedTCB to T is assigned from the LIFO pool of TCB currently maintained on V. The above-mentioned execution context has the highest locality again also here.

[0110]3. When the pool of V is emptynew TCB is assigned from the global pool where this was also constituted in order of LIFO. Each local VP pool is maintaining a number of TCB of threshold tau which it can hold. When a pool overflowsthe VP moves the half of TCB in a local pool to a global pool. When the local pool is not overflowingtau/2TCB is moved to the local pool of VP from a global pool. A global pool plays two roles. minimizing the influence of **** of the program in assignment and the reuse of (1) TCB and (2) -- it is guaranteeing fair distribution of TCB to all the virtual processor. [namely]

[0111]4. Finallywhen TCB cannot be used in any of a global pool or a local poolthe new group of tau/2TCB is generated dynamicallyand is assigned to T. Since both new TCBs are generated only when a global pool and the local pool

of VP are emptythe number of TCBs actually generated in the case of evaluation of the Sting program is collectively determined by all the VP.

[0112]A <virtual processor> virtual processor (extending virtual machine) is a first-class object in a sting. The state of VP called first-class has an important meaning from which either the thread system of a high level and other asynchronous parallel languages distinguish a sting. Parallel computing can be organized by mapping a process clearly in the 1st at a specific virtual processor. For examplethe thread P by which communicating with other threads Q currently performed on VP V closely is known should be performed on VP near V in topology. In Stingsince VP is shown directlysuch consideration is realizable. For exampleit can separate from present VP (for examplepresent VPleft VPright VPupper VPetc.)and the program of a systolic style can be expressed using self-dual addressing. This system provides some default addressing modes to much common topology (for examplea Hypercubea mesha systolic arrayetc.). Since VP can be mapped in a specific physical processorwith the ability to operate a virtual processor as a first-class data value Sting's programA different parallel algorithm defined by various specific processor topology can be expressed very flexibly.

[0113]It explains with reference to the program of drawing 7. This program generates the three-dimensional mesh of the virtual processor multiplexed on the two-dimensional mesh of a physical processor. This array has the heightthe same height as widthand width of a physical machine. By mapping each element of a depth direction in the same virtual processora three-dimensional array is reduced to a two-dimensional array. Thereforethe number of the generated virtual processors is the same as the number of physical processors. All the threads mapped by the processor of the same depth are performed on the same VP. Procedure get-pm-height and get-pm-width are given by a physical machine interface. Addressing is simple referring to the array to the array of a virtual processor which returned only create-3 D-mesh absolutely.

[0114]A create-vp procedure generates new VP which runs on the physical processor which get-pp returned. If topology is generatedit is possible to

separate from present VP and to build a self-dual addressing procedure. For examplethe upper VP procedure which moves to 1 dimension upper part in topology can be defined.

[0115]

(define(up-VP)

(let((address(vp-address(currently-virtual-processor))))

(array-ref 3D-mesh(vector-ref address 0))))

1. Generate the group of the virtual processor mapped by the suitable physical processor.

[0116]2. Relate the address in virtual topology with each VP.

[0117]3. Store a virtual processor in the data structure absolutely used in virtual topology for addressingand define an access routine suitable on the structure.

[0118]4. Define the procedure of self-dual addressing.

[0119]A <thread controller> thread controller treats exchanges with other system elementssuch as a physical processora threadetc. by a virtual processor. The most important function of a thread controller is treating the change state of a thread. When a thread produces the virtual processor which is running on it now by the change statea thread controller certainly decides which thread should be run next by calling a thread policy manager.

[0120]When realizing the thread controller of a stingsome interesting problems become clear. A central change state procedure is shown in drawing 9 and drawing 10. The operation by TCB seen by these procedures cannot be used with user application. Since the thread controller is written in the stingall the synchronous calls to TC procedure are treated as a usual procedure call. Thereforethe activity register which the procedure which is running by the present thread uses is automatically evacuated in TCB of a thread at the time of the entry to a controller.

[0121]Procedure start-context-switch (drawing 8) takes the desirable following state over the present thread (namelythread included in TC) as the argument. The disable of PURIEMPUSHON is carried out first. Nexta new thread (or TCB)

returns by procedure tpm-get-next-thread.

[0122]When there is no thread which can runa procedure returns an imitation (false). In this casethe present thread is run againor procedure tpm-vp-idle is called considering present VP as an argument (noting that it is in a ready state). Procedure tmp-vp-idle can be requested as being able to perform various bookkeeping operations and switching to the physical processor at other VP.

[0123]When the following object is the present TCBno operation is performed but the present thread is resumed promptly. When the returned objects are other TCBsthe state is set as a run and VP field is set as present VP. And the present TCB is recycled in a TCB pool (when the state is DEDDO)the register is evacuatedand the state of new TCB is restored to a processor register.

[0124]In the case of the thread in which the returned object does not have an execution contextTCB is assigned to it. This TCB turns into the present TCBwhen the next-stateno field is DEDDO. Or it is set to TCB assigned from VP local pool or a global pool. A thread starts execution using basic procedure start-new-tcb. This thread applies procedure start-new-thread (refer to drawing 10) using TCB new as that execution context.

[0125]The code (drawing 9) of finish-context-switch is executed by the thread which start-context-switch returned. The lock which the thread (called within this procedure before) by which switch out was carried out in order that that purpose might set up VP field of a new thread holds is releasedIf suitablea former thing will be included in lady queuingand a PURIEMPUSHON time is reset. By including a former thing in queuingonly after a new thread is set up on VPa controller eliminates the race condition between making a change state start and including a thread in lady queuing of VP. Procedure tmp-enqueue-ready-thread and tmp-enqueue-suspended-thread are realized by the thread policy manager.

[0126]The code of start-new-thread is shown in drawing 10. If TCB is assigned to itthe thread object which has thunk E_t can start evaluationand comes to be connected with a default error handler and the suitable Klin rise code. A throw (catch point which start-new-thread sets up) for coming out of E_t is made to

rewind suitable for a thread stackand resourcessuch as a lock which a thread holds by itare released appropriately. The code of recession following evaluation of E_t carries out the garbage collection of the stack and heap of a threadThe value which E_t generated is stored as a part of thread stateall the threads that are waiting for this value are wokenand the new thread which should run is made to choose it as the taele recursive call to a change state procedure. Since E_t is wrapped in dynamic whined formeven when a thread carries out abnormal terminationit is guaranteed that the garbage collection of the storage area of a thread is carried out.

[0127]A garbage collection must be performed before Waiters of a thread is woken up. It is for making the object which is long life and contained the local heap from the thread (the object which the thunk of the thread returned is included) transfer to other activity heaps. It is because other threads will obtain the reference to the storage area of the newly ended thread if this is not performed. Since the storage area of the settled thread is assigned to other threadsthis serves as an error clearly.

[0128]<Thread policy manager> each virtual processor has a thread policy manager. A thread policy manager determines all the policies about the scheduling of the thread on a virtual processorand shift. A thread controller is a thread policy manager's clientand a user's code cannot access it. A thread controller certainly calls a thread policy managerwhen it is necessary to determine in relation to the following thing. Namelywhen the present thread releases a virtual processor for a certain reason with initial mapping to the virtual processor of a threadNextit is which thread the virtual processor should make run and which thread to make transfer from a virtual processor or a virtual processor when.

[0129]Although all the virtual processors have the same thread controllereach virtual processor can be provided with a different policy manager. As for thiseach processor is important especially for the real time application that required scheduling controls a variously different subsystem.

[0130]A thread policy manager provides the interface well defined to the thread controller. The data structure used in order that a thread policy manager may determine is completely private for a thread policy manager. They presuppose that it is local to a specific thread policy manageror various thread policy managers can share themand they can also be made into those combination. Howevernone of other portions of a system can be used. Thereforethe thread policy manager can customize so that different **** to a different virtual machine may be performed. As a resultthe user can customize the determination of a policy according to the kind of program made to run.

[0131]Since VP can be provided with a different thread policy managerrespectivelya different group's thread generated by application can be made into a different object of a scheduling system. A virtual machine or a virtual processor can be adjusted so that a different scheduling protocol or a different scheduling policy may be treated.

[0132]Although the thread controller of a sting defines the change state procedure of a threada transcendental scheduling policyor transcendental load balancing and policy does not give a definition. These policies may be dependent on application. Although some default policies are given as a part of Sting's total execution time environmentthe user can write an own policy freely. In factas shown in drawing 3each virtual processor 16 has the thread policy manager (TPM) 19 of itself. Thereforedifferent VP in the given virtual machine can realize a different policy. TPM19 treats the scheduling of a threadmapping of a processor/threadand shift of a thread.

[0133]It is important for a longevity type parallel (or interactive mode) program to divide application into an individual scheduling group. The thread which performs the procedure relevant to I/O needs different scheduling from the thread which performs the routine relevant to calculation. Application with restrictions of real time should be realized using a different scheduling protocol from what needs only a simple FIFO scheduling policy.

[0134]The parallel program of a tree structure will operate the best by using the

scheduler based on LIFO. The application made to run a master / slave algorithm or a worker firm algorithm will operate to fitness more by using a round robin PURIEMPUSHON scheduler for justice. Since all of such applications are the components of a large program structure object or large program environmentthe pliability acquired by evaluating such applications by a different policy manager is important. The individual application which performs independently the meeting of the thread evaluated on the same virtual machine may exist. The scheduler according to each can have the thread policy manager who has a different performance characteristic and was realized in a different form.

[0135]This invention investigates offer of a flexible framework. And this flexible framework can incorporate a clearly different scheduling system to a userwithout making a change to the thread controller itself. Thereforeall the TPM must follow the same interfacealthough no restrictions are imposed in the realization. The interface shown below chooses the new thread which should runinserts the thread under evaluation in queuingsets up the priority of a threadand provides the operation for making a thread shift. It is for TC to use these procedures exclusively. Generallythe user application does not need to consent to the interface of a thread policy manager and a thread controller.

[0136](tpm-get-next-thread vp) returns the thread of the ready state which should run on vp next.

[0137](tpm-enqueue-ready-thread vp obj) inserts in lady queuing of TPM relevant to vp obj which will be either a thread or TCB.

[0138](tpm-priority priority) And (tmp-quantum quantum) is a guard procedure which checks a priority with each effective argumentor an effective quantum.

[0139](tpm-allocate-vp thread) assigns thread to vp. thread is assigned to the virtual processor become final and conclusive by TPM when vp is an imitation.

[0140](tmp-vp-idle vp) is called by Thread Manager when there is no thread which is evaluating on vp. This procedure can make a thread able to shift from other virtual processorsor can perform bookkeepingor since it has processor switch itself to other VPit can call a physical processor.

[0141](tpm-enqueue-suspend up-thread) suspends thread on the hold queue of vp.

[0142]TPM makes two fundamental decision of load balancing besides determining an order of scheduling over the thread under evaluation. (1) Choose VP which should run the thread generated newly. (2) Decide which thread on VP can be shifted and decide which thread is made to shift from other VP.

[0143]The first determination is important in order to treat early load balancing. The 2nd determination is important in order to support a dynamic load balance and a protocol. A decision of arrangement of the beginning of the thread under evaluation is newly made based on a different priority from the priority used in order to often opt for shift of the thread under present evaluation. A TPM interface saves this distinction.

[0144]A scheduling policy can be classified according to some important matters.

[0145]Locality: Is single global queuing in this systemor does each TPM have queuing of itself?

State: The thread may be distinguished based on those present statesor -for examplea certain applicationmay choose the real overseas subsidiary that all the threads occupy single queuing [be / nothing] with regards to those present states. Or a thread may choose classifying a thread into different queuing based on under evaluationwhether it was scheduledor it is suspended.

[0146]Ordering: Is queuing realized as a structure of FIFOLIFOround robina priorityor real time (in other things)?

Serialization: What kind of locking structure does application impose on various policy managers' queuing?

[0147]A difference arises in the performance characteristic as a result by which choice is chosen by this classification. For examplethe granularity structure distinguished from the thread which had the thread (namelythread which has TCB) under evaluation scheduled is fittedAnd when the restrictions of the ability to make only the scheduled thread shifting are imposedit becomes unnecessary [a lock] accessing queuing of the thread under evaluation. This queuing has

local it to generated VP. Howeversince it is a target of the shift by TPM on other VPqueuing holding the scheduled thread must be locked. This kind of scheduling system is useful when a dynamic load balance is not a problem. Thereforewhen a non-blocking thread (duration time is almost the same) long lasting in a large number existsalmost all VP becomes busy almost all time at execution of the thread on local lady queuing of these selves. Thereforeit is useful to remove the lock on this queuing in such application. The application whichon the other handgenerates the thread to which duration time is changed requires cost in connection with locking lady queuing which can runwhen it uses with TPM which can shift both the scheduled thread and the thread under evaluationbut. Higher performance is shown.

[0148]When a thread policy manager needs to perform a new threadglobal queuing always means competition between thread policy managers. Howeverif such structure is usedin realization of many parallel algorithmsit is useful. For examplein a master / slave (or worker firm) programthe pool of a thread is first generated to a master. These threads are long lasting structures in which these selves do not induce any new threads. These are rarely blockedonce it runs on VP. ThereforeTPM which is performing such a thread does not need to support the overhead of maintaining local thread queuing. Howeverlocal queuing is useful in realization of a parallel programas a result of the structure of a process taking the form of a tree or a graph. These queuing can be used with such applicationin order to load and balance a thread with justice in the group of a virtual processor.

[0149]The <message transmission abstract object> message transmission must be an efficient communication mechanism in the separation memory architecture. In particularthat is right to the coarse parallel application of granularityor the parallel application which has a known communication pattern. A port is the data abstract object established in the stingin order to make the overhead of realizing a shared memory on the separation memory architecture into the minimum thing. A first-class procedure and port show bilateral work in this context.

[0150]A sting makes it possible to unify a message transmission abstract object

in shared memory environment. A port is a first-class data object and is committed as a receptacle to the message sent from other threads. Since a sting uses a share virtual memory modelwith any complex data structures (the closure is included)it leads and it can do a port. For this pliabilitythe application of a sting can realize the message transmission protocol of user levels in a clear formand it becomes possible to combine the strong point which was most excellent in a shared memory and message transmission in the simplified environment.

[0151]A port is a first-class data structure. Two fundamental operations occur to a port.

[0152]1. (put obj port) copies obj to port. This operation is as asynchronous as an informer.

[0153]2. (get port) removes the message of the beginning in portand when port is emptyit blocks it.

[0154]The object read from the port P is a copy of the object written in P. This copy is a shallow copy. That isonly the top structure of the object is copied and the low-ranking structure is shared. Since these ports are designed use when a shared memory is insufficientthey are designed by copying a semantic rule. Although the standard version of put performs a shallow copythere is also a version which performs a deep copy. The version not only copies the top objectbut copies all low-ranking structures.

[0155]For examplethe copy of the closure is built when sending the closure in a message using a shallow copy. Howeverthe reference to the object bundled within the environment which the closure defines is saved. Selection of the copy mechanism to be used is clearly influenced by the physical architecture in back and the field of application. A series of message transmission realization which can be fitted to the specific physical substrate in which the Sting realization exists exists.

[0156]Thereforeevaluation (put(lambda()E) port) of the following expression The closure of a procedure (lambda()E) is sent out to port. If the receiver is defined as follows on port (define (receiver port))

(let((msg(get port)))
(fork-thread(msg)(current-vp))
(receiver)))
The sent-out procedure is evaluated on this receiver's virtual processor. The receiver can accept a new request in parallel to processing of an old request by generating a new threadin order to evaluate a message.
[0157]Communication of this style is called the "active message." That is because the operation which should be performed when a message is received is not coded as some basic systems but message itself opts for it. Since the interface of a virtual processor and a thread does not need any change in order to support message communicationvery large pliability and simplicity are obtained with such a model. Two things in the design of a sting are important for realization of this function. (1) Since an object exists in share virtual memoryall the objects (the object which has a REFA lance to other objectsfor examplethe closureis included) can be freely transmitted between virtual processors. (2) A first-class procedure enables construction of the complicated message handler which a user defines. These handlers can be performed within the separated thread on one of virtual processors. In a separation memory machinean object will exist in distributed common virtual memories. E is considered as a complicated inquiry of a database in an above-mentioned example for explanation. Supposing a receiver is illustrated on the processor in which this database existssuch an inquiry will not be accompanied by the shift which a database's own cost requires. Since an inquiry is directly copied to the processor in which a database existscommunicative cost is reduced. It is not necessary to shift to the processor which performs an inquiry in itself [database]. With the capability to send a procedure to the data instead of communication of a more traditional RPC styleimprovement in important performance and expression nature may be obtained in some respects.
[0158]A first-class procedure and lightweight thread transmit the communication abstract object of an attractive high level in an active message. In the system

which supports an active message without using these abstract objectsthis function is typically realized by the low support protocol. A first-class procedure makes it possible to realize an active message commonly. An active message is a procedure sent to a port. A first-class port has a clear and important use also in distributed calculation environmentand enables realization of a programming model easy and cleaner than the conventional PRC.

[0159]A <memory management> sting uses a share virtual memory model. On a dispersion memory plat formSting must be built on a distributed-common-virtual-memories substrate. Thereforeit does not depend for the meaning of referencewithout where reference is generated or where an object being physically.

[0160]In <memory-mechanisms> Stingthere are three storage areas in relation to each TCB32 (drawing 5). The 1st is the stack 31 and is used for assignment of the object generated by the thread. Although the life of this thread generated itit does not exceed the dynamic range. Refer only to other objects assigned to the present stack frame (or before) or other objects which were assigned to the heap for the object assigned on the stack more correctly. Refer to the object in a heap for the object to which the stack was assigned. The thread relevant to the stack is because it becomes suspensionwhile the garbage collection of the heap 33 is carried out. The reference information contained in the stack is a part of route which is said to have traced with the garbage collector.

[0161]The private heap 33i.e.a local heapis used to the assigned non-common object for a thread. This object may exceed the life of the procedure in which that life generated the object. It was presupposed that it may exceed because a compiler was not always able to determine the life of an object in programming languagessuch as a scheme and ML. The life of an object may not be decided in the language which can be called to a strange procedure. Although the reference information contained in the private heap can show other objects in the same private heapor the share heap 35i.e.a global heapthe object in the stack 31 cannot be shown. Although the reference information in this stack can show the

object in a private heapthe reference information in a share heap cannot show these objects. Since the data assigned to the private heap is exclusively used by the single thread of controla higher locality realizes it by a private heap. The object near mutually [that there is no assignment inserted among two or more threads] within a heap means that a possibility of being related mutually logically is high.

[0162]Other threads cannot access the object contained in the stack of a threador the local heap. Thereforeboth the stacks and local heaps of a thread can be realized in the local memory on a processorwithout taking the coherency of synchronization or a memory into consideration. The local heaps of a thread are a series of heaps organized generationally actually. In the youngest generationassignment of a storage area is always performed like other generational collectors. An object is moved to an old generation as age becomes high. All the garbage collections of a local heap are performed by thread itself. In almost all the thread system that supports a garbage collectionall the threads in a system must be suspended between garbage collections. The thread of a sting carries out the garbage collection of those local heaps to other threads independently and in asynchronous to it. Thereforeother threads can continue calculationwhile the specific thread collects the local heap. As a resultmore outstanding load balancing and a high throughput are obtained. The 2nd strong point of the strategy of this garbage collection is at the point that the cost concerning the garbage collection of a local heap is imposed on not all the threads in a systembut is imposed only on the thread which assigns a storage area.

[0163]A sting gives "thread groups" as a means for controlling the meeting of a related thread. A child's thread belongs to the same group as the parentsunless it was generated as some new groups. Thread groups with the operations (for examplereport etc. of the listing of all the threads in the given groupall the groups' listingprofilingand a system) for debugging and a monitor. The operations (for examplean endsuspensionetc.) of the usual thread are given. Thread groups

contain again the "share heap" which all the members can access.
[0164]When thread groups are generatedthe share heap 35 of thread groupsi.e.a global heapis assigned. Share heaps like a local heap are a series of heaps organized generationally actually. The reference information in a share heap can show only the object in a share heap. All the objects by which this is referred to from a common object are common objects.
Thereforeit is because it must exist in a share heap.
The restrictions to this share heap are in (a) share heapor are assigned in the (b) local heapIt carries out by guaranteeing that the reference information memorized by the share heap directs the object by which the garbage collection is carried out into the share heap. That isthe graph of the object which can reach from the object referred to must be copied or arranged in a share heap. The overhead of this memory model is decided by frequently [how much] the reference information to the object assigned on the local heap escapes. According to experiencewhen realizing a fine grained parallel programto a related threadmost objects assigned to the local heap are localand are not shared continuously. The object frequently shared between threads is easily detected by analysis of a language abstract object or compile time.
[0165]When it summarizesthe reference order between the thread fields relevant to a certain thread is as follows. The reference information in (1) stack Namelythe stack frame of the present or beforeOr the object in a local heap or a share heap is shownand the reference information in (2) local heapsThe object assigned to the object or a certain share heap on the heap is shownand the reference information in (3) share heap shows the object assigned to the share heap (or other share heaps of a certain).
[0166]Like a local heapalthough the global heap is organized generationallyits garbage collection of a global heap is more complicated than the thing to a local heap. That is because the thread from which a large number differ may access the object in a global heap simultaneously. As a resultthe lock of a heap is required for assignment of a global heap.

[0167]In order to carry out the garbage collection of the global heapall the threads (and thing of the low rank) in related thread groups are suspended. That is because all of these threads can access the data in a global heap. Howeverwhat is not into other threads in a systemi.e.the group relevant to the heap by which a garbage collection is carried outcontinues execution regardless of a garbage collection.

[0168]Each global heap has the reference information which comes there in relation to it. These groups are maintained by the check to memory of the reference information which crosses an area boundary. After the thread relevant to a global heap is suspendeda garbage collector uses the group of arrival reference information as a route for a garbage collection. All the objects that can reach from the group of arrival reference information are copied to a new heap. An end of a garbage collection will resume the thread relevant to a global heap.

[0169]<An abstract physical machine and abstract physical processor> The abstract object of a low is micro kernel of this operating system called an abstract physical machine (APM) most.

[0170]APM plays three important roles in sting software architecture.

[0171]1. Provide the safe and efficient foundation which supports two or more virtual machines.

[0172]2. Separate other elements of all the in a system from the feature and singularity depending on hardware.

[0173]3. Control access to the physical hardware of a system.

[0174]APM is realized within the special virtual machine called a route virtual machine. This machine has an accessing means for the function which can be used with any of other virtual machines containing virtual address spacea virtual processorand a thread. The route virtual machine has an accessing means for an abstract physical processora device driverand a virtual memory manager. The abstract physical machine is constituted by the virtual machine andas a resultsome important expression nature is obtained. There is no heavyweight thread. All the threads are lightweight. There is no kernel thread or stack which

realizes a system call. All the system calls are treated using the execution context of the thread which creates a system call. This is a language with a safe scheme (that isthe free compulsion between a dangling pointerand an address and data is impossible)and since the portion of APM is mapped by all the virtual machines in a systemit is possible. The thread in APM can also use the asynchronous programming building body which a user's thread can use. The thread relevant to APM is controllable like other threads of all the in a virtual machine. The thread which blocks execution of kernel operation notifies that to the virtual processor of these threads. VP can perform other threads of a certain freely by it. In the case of both the communication between threadsand I/Othis is performed. A sting processes non-blocking Carnell Kohl so that the same capability as starting of a scheduler or the virtual processor abstract object of Psyche may be provided for example.

[0175]A virtual machine is generated by APM and destroyed. The following things follow on generation of a new virtual machine.

[0176]1. Generate new virtual address space.

[0177]2. Map an APM kernel in this address space.

[0178]3. Generate a route virtual processor in this virtual mapping.

[0179]4. Assign an abstract physical processor to this mapping.

[0180]5. In order to make it run on an abstract physical processorschedule the above-mentioned route virtual processor.

[0181]The signal for making destruction of a virtual machine end all the threads which are running on the machine is generatedAll the devices that the thread currently performed within a machine opened are closedand canceling assignment of the virtual address space relevant to this machine at the last follows.

[0182]Each processor abstract object 12 comprises the virtual processor controller (VPC) 13 and the virtual processor policy manager (VPPM) 15. The relation between VP controller and VP policy manager is as of the same kind as the relation between a thread controller and a thread policy managernamelyVP

controller is VP policy manager's client. When it is necessary for VP controller to determine a policyVP controller certainly calls VP policy managerin order to make the decision.

[0183]Although all physical processors make the same VP controller runthey can make a different VP policy manager run. As a resulta multiprocessor system becomes possible [customizing use of each physical processor by a system]. As for a systemit is possible to also make the same VP policy manager run on each physical processor.

[0184]When a virtual machine tends to schedule the virtual processor on an abstract physical processora virtual machine calls the virtual processor controller on the physical processor. Similarlywhen a virtual machine tends to remove a virtual processor from an abstract physical processora virtual machine calls the virtual processor controller on the physical processor. Each VP controller manages the virtual processors including the change of state of a virtual processor mapped by the physical processor.

[0185]VP policy manager determines all the policies concerning the scheduling of the virtual processor on a physical processorand shift. There are three types of these determination. VP policy manager decides mapping to PP from VP to be the 1st. This mapping is performed to two different timing. That isthey are a time of VP running firstand a time of blocked VP running again. A policy manager determines the turn and the period which are made to run VP on PP as the 2nd. FinallyVP policy manager opts for when VP should be moved to other processors from a certain processor (shift).

[0186]By these three determinationVP policy manager can balance the work load on a machineand can decide the character concerning the justice of the physical machine about a virtual machine. When a physical processor breaks downit can opt for where VP of fault-tolerant VM is moved.

[0187]VP provides the interface well defined to VP controller like a thread policy manager. The data structure used in order that VP policy manager may make the decision is completely private life to VP policy manager. To specific VP policy

managerthese data structures are made as it is localor in VP policy manager's versatilitycan be sharedor are made with those combination. Howeverother elements of a system do not have an accessing means for them. VP policy manager can customize so that different **** to the case where Sting differs may be carried out. It becomes possible to customize a sting with this function to various operating system environmentsuch as a real-time systeman interactive mode systemand a system that performs a lot of calculations.

[0188]Finallyalthough the thread policy manager is involved in load balancing and justice between threadsthe virtual processor policy manager is involved in load balancing and justice between virtual machines and between virtual processors.

[0189]Each physical processor in APM contains the virtual processor controller (VPC) and the virtual processor policy manager (VPPM). The physical processor is structurally the same as that of a virtual processor at this point. VPC affects the change of state on a virtual processor. Like a threadthe virtual processor can take the state of a runa ladya block or an end either. VP of the run condition is performed on the physical processor now. Although VP of a ready state can be runit is not running now. VP of block status is performing the thread which is waiting for a certain external event (for exampleI/O). VPPM performs scheduling of VP on a physical processor. The scheduling policy is the same as that of what TPM uses. VPPM provides the interface defined well to VP controller. Different Sting's system can be provided with a different VP policy manager.

[0190]The exception and interruption of which the <treatment of exceptional> synchronization was done are uniformly treated by a sting. The handler which performs 1 set of operations for treating an exception relates to all the exceptions. A handler is a procedure performed within a thread. The exception produced on the processor P is performed using the context of the present thread of P. There is no special exception stack into the micro kernel of a sting. When the exceptions (for exampleinvalid command and memory protection destruction etc.) which they are on the processor P arisethe present continuations (namelya

program countera heap frontiera stacketc.) of P are evacuated first. Nextin order that an exception dispatcher may find an exceptional targetwhen the thread is runninghe interrupts itand pushes the continuation and argument of a handler on the stack of a target thread. Next. [whether a dispatcher makes the thread of (a) present resume by making it return to it simplyand] (b) or it makes a target thread resume -- or (c) -- in order to make other ones on this processor of threads resumecall a thread controller or choose either. When a target thread is resumedthe thread performs the continuation of the top of the stack. This is a continuation of an exception handler.

[0191]This exception-handling means in a sting is excellent in some respects.

[0192]1. Since this exception-handling means is a procedurean exception can be treated only by calling it.

[0193]2. An exception is treated in the execution context of the thread which receives an execution context.

[0194]3. An exception is dispatched in the context of the present thread.

[0195]4. The exception dispatched once serves as the present continuation of a target threadand when a thread is resumedit is performed automatically.

[0196]5. An exception is treated only when a target thread is resumed.

[0197]6. the code treating an exception -- a scheme -- ******** -- he and its code operate a continuation and a procedureand attain a desired effect.

[0198]A first-class procedurea threada clear continuationAll of assignment of a dynamic storage area and a uniform addressing mechanism are the central features of a design of a stingandas a resultthe sting can give the model of this exception.

[0199]The target thread of the exception which synchronized is always the present thread. An asynchronous exceptioni.e.interruptionis treated in a slightly different form. Since interruption is controllable by every thread (it is not a thread under present execution)in order to treat such an exceptionthe handler needs to interrupt the thread which processes an exception directly or is running nowand needs to treat an exceptionor needs to generate a new handler. What is

necessary is just to set up the only suitable present continuation of a thread call a handlerwhen establishing a thread or using the present threadin order to perform a handler since an interrupt handler is also a procedure of a scheme. The pseudocode for the exception dispatcher of a sting is shown below.

[0200]

```
1:(define(exception-dispatchertype.
args)
2: (save-current-continuation)
3: (let((target handler(get-target&handler type args)))
4: (cond((eq?target(current-thread))
5: (apply handler args))
6: (else 7: (signal target handler args)
8: (case((exception-priority type))
9: ((continue)(return))
10: ((immediate)(switch-to-thread target))
11; ((reschedule)(yield-processor)))))))
```

In the line 2the present continuation is evacuated to the stack of the present thread. Since this continuation cannot be escaped and is called only onceit can evacuate the above-mentioned continuation to the above-mentioned stack. In the line 3a dispatcher finds the thread which is the exceptional targetand the handler to an exceptional type. In the line 4if it is confirmed whether an exceptional target is the present thread and it meetsan exception continuation will not be pushed (line 5). A dispatcher applies a handler to the argument simply rather. Since the dispatcher has already run in the context of the exception target (namelythe present thread)this [his] is effective. When an exceptional target is not the present threada dispatcher sends an exception to a target thread (line 7). It is equivalent to pushing the continuation which interrupts a thread and contains a signal handler and its argument to the stack of a threadand making the thread which a signal handler is made to perform resume to send a signal to a thread. After sending a signal to a target threada handler decides which thread to run

next on a processor (line 8). Itself may be running (line 9) or it also has a case of a target thread (line 10) or a thread with the highest priority (line 11).

[0201]It differs at the point with one more important exception handling function of a sting and thing in other operating systems. Since an exception handler is the usual first-class procedure since the thread treating an exception does not differ from the thread of the user levels in a system (for examplethey have an own stack and heap) andThe handler can assign a storage area freely. The data generated by the handler is the same method as other data being restoredand reclaimed rubber will be carried out by a garbage collector. The mechanism of exception handlingand when a device driver is realized more for the homogeneity between the abstract objects of the sting of a high levelhigh expression nature and high efficiency are obtained. In a parallel language or a parallel operating systemthis is not realizedwhen there is no above-mentioned homogeneity.

[0202]Since all of a first-class procedureassignment of a threada clear continuationand a dynamic storage areaand a uniform addressing mechanism are the features of Sting's designSting can give the model of this exception.

[0203]Although more than the <parallel paradigm> explained software architecture in detailbelow some extensive examples explain a parallel paradigmand the software architecture of this invention realizes it.

[0204]In the concurrent program as a resulteach process performed in parallel affects the value of complex data structure (for examplean array or a list). Or each process is a member of the graph of a complex process. Communication of a process is based on the structure or graph of this result. Expression which tries access to the element of the result which the contributing program is still evaluating is blocked until a program is completed. A future is a good example of the operation which was dramatically suitable for carrying out the parallel algorithm as a result. The object (future E) generated by expression of MultiLisp or Mul-T generates the thread for the calculation E. And the object by which the return was carried out is known as a future. When v is produced as a result and E is completedit is said that the future was become final and conclusive.

Expression which touches a future is blocked when E is still calculatedand v is given when another side and a future are established.

[0205]In the simple sorting program shown in drawing 11each example of a future is accompanied by generation of a new thread. This **** is not desirable. The future in which it calculates on the level i of a process tree is based on the reason for having clear data dependency to the child in the level i+1 etc. When there is data dependency in this programa result to which the availability of a processor and memory storage falls is brought. Many of generated lightweight processes this. Compared with cost required in the case of the process of blocking when still requesting other values as a thing of an unevaluated futureor calculating a small prime numberfor examplein order to generate themit is for performing a little calculations.

[0206]Since the dynamic state of a thread comprises a large object (for examplea stack and a heap)when blocking of a process arises frequentlyor when granularity is too small in a processa compromise is reached about the locality which are cash and a page.

[0207]The future F by which the semantic rule of a touch and a future touches other futures G orders to have to block by Gwhen G has not been become final and conclusive yet. T_F and T_G are considered as thread expression of F and Grespectively. (a) delay or when it is scheduled [TB]while carrying out (b) evaluation of the run time dynamics of the touch operation in Gin either (c) or when [it becomes final and conclusive] they may be accompanied by access to T_G. In the last casesynchronization is unnecessary among these threads. In the case of a case (b)it is necessary to block T_F until T_G is completed. In the case of a case (a)important optimization is performed by a sting. This is explained below.

[0208]TF can evaluate rather the closure (referred to as E) shut up in TG using the stacked heap of itself rather than blocking and forcing a context switch. By a stingE is treated as a usual procedure and the touch of G is actually treated as a simple procedure call. In this caseit is said that T_F absorbs T_G. This optimization is the right at the point of blocking T_F inevitably in the case of others. By applying

E using the dynamic context of T_FVP in which the T_F operates does not undertake the overhead of performing a context switch. Since TCB of T_F is used insteadit is not necessary to assign TCB to T_G.

[0209]This optimization may become drawing the result of having been conspicuous and differedwhen the called thread does not necessarily need to block and it is used. For exampleT_G presupposes that it was an element of the speculative call by T_F. Although T_G branchesother speculative threads (it is called T_H) presuppose that it does not branch. When there is no absorptionboth T_G and T_H induce a separate thread context. Howeverwhen there is absorptionsince T_F can absorb T_G and T_G carries out a loopit will carry out the loop also of the T_F. When a thread can be absorbed or it cannot doa user parameterizes the state of a thread and can notify to TC. A sting provides the interface procedure for it.

[0210]A sting reduces the overhead of context SUITCHIINGU for absorptionand the granularity [as opposed to / when a process shows strong data dependency mutually in a program / the program] of a process is increased. Of coursein order to make operation the most effectivethe granularity of a thread must be large enough so that the scheduled thread can be absorbed. When the granularity of a process is too smallbefore the absorbed thread can require those valuesa processor will start evaluation of the thread which may be absorbed.

[0211]The yne lining and the lazy creation of task based on load are the optimization of two others of the same kind applied to other parallel Lisp systems. In the yne lining based on loadwhen a specific threshold with the load of the present system is crossedin-line (namelyabsorption) one of the thread is carried out. Not only a programmer's intervention may be unnecessarybut by this optimizationthe program which should originally be ended may be in a deadlock or a prolonged halt condition under a certain kind of conditions. This is because determination of yne lining cannot be withdrawn. Thereforein this optimizationwhen a task needs to be estimated by the order which exists for that data dependencyan order of different specific evaluation from it is imposed on a task. Since absorption of a thread is produced only when are not absorbed and a

thread blocksand only when the dependency of data is guaranteedit is not influenced by this problem.

[0212]A lazy creation of task solves many problems concerning the yne lining based on load. In a lazy creation of taskalthough in-line ones of the evaluation of all the threads is always carried outhowever when a processor is in an idle statewithdrawal of this yne lining operation is enabled. A thread is never generatedunless it is actually needed. In this designa programmer's intervention is not neededand the deadlock of the program which originally does not carry out a deadlock is not causedbut the number of the tasks actually generated is reduced.

[0213]Absorption of the thread mainly differs from the lazy task at two points. (1) Even if the scheduling protocol decided with application existscommit absorption of a thread. Lazy creation of task assumes a global LIFO schedule and existence of single queuing for holding the thread by which in-line one was carried out. (2) One global heap is used for lazy creation of task to one processor. In a lazy creation of taskwhen the steel of the task is carried outa locality falls from the case where it is thread absorption. It is necessary to make the 2nd stop all the threads in a system in the garbage collection in the case of lazy creation of task (even if collector itself is parallel). In absorption of a threadthese restrictions do not exist.

[0214]Other examples are the paradigms of a master slave and this is the popular art for constituting a parallel program. In this artthe collection of the generated process is performed transcendentally. A master process generates some worker processes and combines those results. Communication of a process is typically performed through a share parallel data structure or a share parallel variable. A master slave program is often more efficient than the parallel program of the result on a stock multiprocessor plat form. That is because a worker does not almost need to communicate mutually except for the case where those results are published. And the granularity of a process can be adjusted and higher performance is obtained.

[0215]The sting was usedin order to optimize the first-class tuple space in a scheme and to realize. tuple space -- synchronization content AKUSE -- sable -- it is an object which functions as an abstract object of - memory. Tuple space is natural selection for materializing the algorithm of a large number based on a master / slave.

[0216]A tuple is an objecttuple operation is binding expressionand since it is not a statementmodularity and expression nature improve further by the existence of tuple space in which first-class directions are possible. In a desirable exampletuple space can be specialized as the synchronized vectorqueuingthe streamthe setthe share variablesignalor bag. The operation permitted on tuple space is eternal in those displays. If application is requiredthe succession class between tuple space can be specified.

[0217]The process can read a new tuple into tuple spacecan remove itor can be deposited. The tuple and argument in reading operation or removal operation are called a "template"and can include the variable attached with "? near at hand." Such a variable is called as it is "formal"and it gains a binding value as a result of match operation. The binding value gained with these full dresses is used in evaluation of a low-ranking expression. Thereforeit can write as follows.

[0218](get TS[?x]

(put TS[(+x1]))

One tuple is removed from TS by thisand it **********s only 1and it is again deposited with TS.

[0219]In this exampleconstruction of the fine parallel program of the granularity which also uses thread absorption and synchronizes on tuple space again is enabled. A thread is used as a genuine element within a tuple. The process P of performing the following expression is considered.

[0220](rd TS[x1 x2]E)

Herex1 and x2 are non-full dresses. It is assumed that the tuple in TS is deposited as a result of operation (spaun TS [E_1E_2]). This operation schedules two threads (it is called TE_1 and TE_2) which calculate E_1 and E_2. If both TE_1 and

TE_2 are completedthe tuple as a result contains two settled threads. A matching procedure applies thread-valuewhen a thread is encountered within a tuple. This operation collects the values of that thread.

[0221]Howeverwhen P is performed and TE_1 is still scheduledP can absorb it freelyand when the result is in agreement with x1it becomes final and conclusive it. When a match does not existP leaves TE_2 which may be in the state where it was scheduledand follows it to the search of other tuples. ThenTE_2 will be absorbedif other processes are able to investigate this same tuple and there is a legal excuse. Similarlyif the result of TE_1 is in agreement with x1P can absorb TE_2 freely next. When either TE_1 or TE_2 has already evaluatedP chooses whether it blocks by one thread (or both)or the tuple which may otherwise be in agreement is investigated within TS. The semantic rule of tuple space does not impose restrictions to this example at this point.

[0222]The combination of the first-class thread and thread absorption of a sting makes it possible to write the fine (result) parallel program of the granularity driven by false demand using shared data structure. In this meaninga thread system tries minimization of the meaningful distinction between the synchronization (for exampletuple space) based on structureand synchronization (for examplea future/touch) of a data flow style.

[0223]Although speculative parallel is important programming techniqueit cannot be used often effectively because of the overhead of the run time produced when it is realized. Two features most frequently involved in the system which supports a speculative programming model are the capability encouraging a task much more more promising than other thingsand having a means which stops and reuses unnecessary calculation (probably cancellation [And]).

[0224]By the following thinga sting enables a programmer to write speculative application.

[0225]1. A user is enabled to program the priority of a thread clearly.

[0226]2. When other threads are completedit can be made to carry out weight of a certain thread.

[0227]3. A thread makes it possible to terminate other threads.

[0228]Since a priority is programmablea promising task can be performed ahead of a thing without that right. Which can wake the thread by which the task alpha first ended in the group of a task is blocked in the case of the end. With this functionthe sting can support the useful gestalt of OR parallel. The task alpha can terminate all other tasks in the group of the taskif it becomes final and conclusive that those results are unnecessary. Howeverthe theoretical estimation using Sting could not cancel the nonlocal side effects brought about by the unnecessary task. The mechanism of reversion with this fundamental system is not provided.

[0229]It considers realizing a wait-for-one construct. This operator evaluates the list of this argument in parallelreturns the value generated by that first argumentand is ended. Thereforeif this expression returns v and a programmer needs it when v arises from a_1 in expression (wait-for-one a_1a_2 ... a_1 ... a_n)All the a_j which remainsand evaluation of j!=1 are ended.

[0230]The specification of the wait-for-all construct which realized AND parallel is the sameand it is. This also evaluates the argument in parallel. Howevertruth is returned only when all the arguments are finished. Thereforesince the thread which performs this expression is blocked until all the a_1 finishesexpression (wait-for-all a_1a_2 ... a_1 ... a_n) functions as a barrier synchronization point. Realization of this operation resembles realization of speculative wait-for-one operation dramatically.

[0231]TC realizes such operations using block-onset which is a common procedure. The thread and TCB are defined as supporting this function. For examplethe information about the number of the threads in the group who ends before being able to resume the thread to which TCB relates which will not become if it kicks relates to the TCB structure.

[0232]block-on-set takes the list and count of a thread. These threads support the argument of the wait-for-one operation mentioned above and wait-for-all operation. The argument of the count expresses the number of the threads which

must be ended before the present thread (namelythread which is performing block-on-set) can accept resumption. When this number is 1a result realizes wait-for-oneand when the number of the above is na result is realization of wait-for-all.

[0233]The relation between thread T_g in a group and the present thread (T_w) which should wait for T is maintained within a data structure (called the thread barrier (TB)) including the reference to the following.

[0234]1. The program which defines other Waiters' TB(when it exists) block-on-set blocked on TCB2.T_g of T_w is shown in drawing 12.

[0235]The next call (block-on-set m T_1T_2 ... T_n)

An amplifier lock is carried out when m T_1 (m<=n) is completed to the thread (it is called T) of ******. Each of these T_1 has the reference to T in those Waiters' chain.

[0236]Application is used with procedure wakeup-waiters started by $a_1$when application ends bkock-on-set. wakeup-waiters investigates the list of Waiters of the letter of a chain from the Waiters slot in the thread argument. Waiters from whom the number of weight becomes zero is inserted in lady queuing of one of VP. TC always starts wakeup-waiterswhen the thread T is completed (when T is completedor when it exists unusuallyit is always). All the threads that are waiting for the end of T are carried out in this wayand are rescheduled.

[0237]If two procedures are given to thiswait-for-one can be defined briefly as follows.

[0238]

(define(wait-for-one . block-group)

(block-on-group 1 block-group)

(map thread-terminate block-group)

When T performs wait-for-oneit is blocked on all the threads in a block-group argument. When T is resumedT is arranged at lady queuing in TPM of one which can be used of virtual processors. The map procedure performed at the time of resumption of T terminates all the threads in the group.

[0239]Procedure wait-for-all of a sting can omit this operation. That is because

ending all the threads in that block group before the thread which performs this operation is resumed is guaranteed.

[0240]Sting was realized in both Silicon Graphics Power Series (MIPS R3000) of eight processorsand Silicon Graphics Challenge (MIPS R4400) of 16 processors. both machines are share (cash - coherent) multiprocessors. A physical processor is mapped by the lightweight Unix thread in this abstract physical machine composition. Each processor in a machine makes one of such the threads run.

[0241]As mentioned abovealthough the desirable example of the computer software architecture was described and being explainedit is possible to add various modification and changewithout deviating from the extensive principle and meaning of this invention so that clearly for a person skilled in the art.

[0242]

[Effect of the Invention]As explained abovein order to control an advanced parallel multiprocessor / multi-computer system according to this inventionThe control system of the advanced parallel computer system using the operating system architecture of the computer which serves as an efficient substrate very much to a present-day programming language is obtained.

[0243]According to this inventionthe control system of the advanced parallel computer system using the software architecture for the asynchronous calculation based on a customizable virtual machine is obtained.

[0244]According to this inventionthe control system of the advanced parallel computer system using the software architecture which supports a lightweight thread as a first-class object on a virtual processor is obtained.

[0245]According to this inventionthe control system of the advanced parallel computer system using the software architecture which contains especially in user levels the policy manager who can customize is obtained.

[0246]According to this inventionthe control system of the advanced parallel computer system using the software architecture containing customizable virtual topology is obtained.

[0247]According to this inventionthe control system of the advanced parallel

computer system using the software architecture containing the thread groups as a place thread absorptiondelay TCB assignmentand memory storage shared is obtained.

[0248]According to this inventionthe control system of the advanced parallel computer system using software architecture including the port of various gestalten is obtained.

[0249]According to this inventionthe computer systems controlled using the above software architecture are obtained.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a block diagram showing the control system using the software architecture by one example of this invention.

[Drawing 2]It is a figure showing the abstract physical machine and virtual machine of drawing 1.

[Drawing 3]It is a schematic block diagram showing the abstract architecture of the operating system of this invention.

[Drawing 4]It is a figure showing the transition of the state of a threadand the state of TCB used by this invention.

[Drawing 5]It is a schematic diagram showing the composition of the memory storage used by this invention.

[Drawing 6]It is a figure showing the program for explaining programming of the thread used by this invention.

[Drawing 7]It is a figure for explaining the program which generates 3D mesh of the virtual processor multiplexed on 2D mesh of the physical processor used by this invention.

[Drawing 8]It is a figure showing the program which puts into operation the context switch used by this invention.

[Drawing 9]It is a figure showing the program which ends the context switch used by this invention.

[Drawing 10]It is a figure showing the program which starts the new switch used by this invention.

[Drawing 11]It is a figure showing the program of the top procedure for the fine adaptation parallel sort algorithm of the granularity used by this invention.

[Drawing 12]It is a figure showing the program which defines block-on-set used by this invention.

[Description of Notations]

10 Abstract physical machine

11 Physical topology

12 Abstract physical processor

13 Virtual processor controller

14 Virtual machine

15 Virtual processor policy manager

16 Virtual processor

17 Thread controller

18 Thread

19 Thread policy manager

20 20' Virtual topology

24 A virtual machine/address space

26 Global storage pool

28 Global common object

30 Route environment

31 Stack

32 TCB

33 Local heap

35 Global heap

36 Delay

38 Schedule